

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2000259445 A

(43) Date of publication of application: 22.09.00

(51) Int. Cl

G06F 11/26

G06F 11/28

(21) Application number: 11058248

(71) Applicant: NEC CORP

(22) Date of filing: 05.03.99

(72) Inventor: IKEDA CHINATSU  
TOMIMITSU KOJI

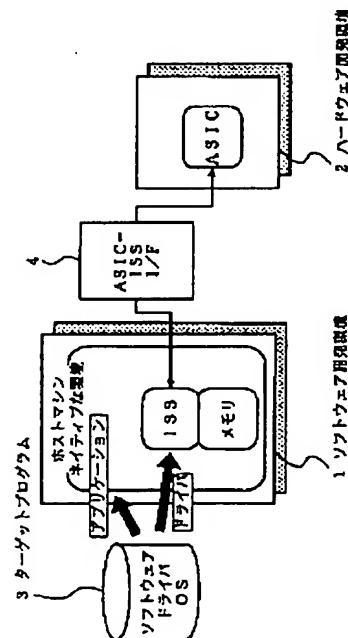
(54) COOPERATIVE SOFTWARE/HARDWARE  
SIMULATION METHOD

(57) Abstract:

PROBLEM TO BE SOLVED: To enable high-speed exact simulation by using an Instruction Set Simulator(ISS) for the part nearer to a hardware and using a host machine native environment for an application to be run by a CPU.

SOLUTION: In a hardware development environment 2, the LSI (ASIC) of a part other than a target CPU is simulated and verified and in a software development environment 1, the hardware at the part of the target CPU is simulated by the ISS and verified. In a program to be run by the target CPU, an OS or device driver for directly operating the hardware is compiled for the target CPU and since there is no hardware dependency and simulation accuracy is not required, the application to be run in spite of the hardware is compiled in host machine native state and run on a host.

COPYRIGHT: (C)2000,JPO



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-259445

(P2000-259445A)

(43)公開日 平成12年9月22日(2000.9.22)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 11/26  
11/28

識別記号  
3 4 0

F I  
G 0 6 F 11/26  
11/28

テマコード(参考)  
5 B 0 4 2  
3 4 0 C 5 B 0 4 8

審査請求 有 請求項の数10 O.L (全 6 頁)

(21)出願番号 特願平11-58248

(22)出願日 平成11年3月5日(1999.3.5)

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 池田 千夏

東京都港区芝五丁目7番1号 日本電気株式会社内

(72)発明者 富満 康治

東京都港区芝五丁目7番1号 日本電気株式会社内

(74)代理人 100086645

弁理士 岩佐 義幸

Fターム(参考) 5B042 GA21 HH07

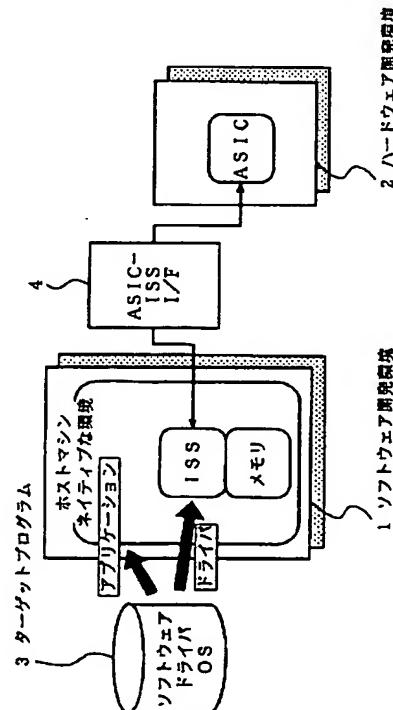
5B048 AA20 DD14

(54)【発明の名称】 ソフトウェア/ハードウェア協調シミュレーション方法

(57)【要約】

【課題】 高速かつ正確にシミュレーションを行う。

【解決手段】 C P U以外のハードウェアはハードウェア記述言語を用いてシミュレーションを行い、ハードウェアを直接制御するソフトウェアはC P Uシミュレータを用いてシミュレーションを行い、ハードウェアを直接制御しないソフトウェアはホストネイティブでシミュレーションを行う。



【特許請求の範囲】

【請求項1】ハードウェアを直接制御するソフトウェアはCPUの命令セットを用いたシミュレータを用いてシミュレーションを行い、ハードウェアを直接制御しないソフトウェアはホストネイティブでシミュレーションを行うことを特徴とするシミュレーション方法。

【請求項2】前記CPUのみアクセスするメモリを、前記シミュレータ側にローカルに持つことを特徴とする請求項1に記載のシミュレーション方法。

【請求項3】前記シミュレータにインストラクション・セット・シミュレータを用いることを特徴とする請求項1または2に記載のシミュレーション方法。

【請求項4】前記ハードウェアを直接制御するソフトウェアには、デバイスドライバ、OSを含むことを特徴とする請求項1～3のいずれかに記載のシミュレーション方法。

【請求項5】CPU以外のハードウェアはハードウェア記述言語を用いてシミュレーションを行い、ハードウェアを直接制御するソフトウェアはCPUの命令セットを用いたシミュレータを用いてシミュレーションを行い、ハードウェアを直接制御しないソフトウェアはホストネイティブでシミュレーションを行うことを特徴とするソフトウェア／ハードウェア協調シミュレーション方法。

【請求項6】前記CPUのみアクセスするメモリを、前記シミュレータ側にローカルに持つことを特徴とする請求項5に記載のソフトウェア／ハードウェア協調シミュレーション方法。

【請求項7】前記シミュレータにインストラクション・セット・シミュレータを用いることを特徴とする請求項5または6に記載のソフトウェア／ハードウェア協調シミュレーション方法。

【請求項8】前記ハードウェアを直接制御するソフトウェアには、デバイスドライバ、OSを含むことを特徴とする請求項5～7のいずれかに記載のソフトウェア／ハードウェア協調シミュレーション方法。

【請求項9】ソフトウェアシミュレータを含む論理シミュレーション装置であって、実行プログラムが実行される際のプログラムアドレスに対応して実行プログラムの命令コードの実行環境情報を記憶する手段と、プログラムカウンタの示すプログラムアドレスから命令コードの実行環境情報を前記記憶手段より読み出して当該命令コードの実行環境を切り替える手段とを有することを特徴とする論理シミュレーション装置。

【請求項10】ソフトウェアシミュレータを含む論理シミュレーション方法であって、実行プログラムが実行される際のプログラムアドレスに対応して実行プログラムの命令コードの実行環境情報を記憶手段に記憶し、プログラムカウンタの示すプログラムアドレスから命令コードの実行環境情報を前記記憶手段より読み出して当該命令コードの実行環境を切り替えることを特徴とする論理

シミュレーション方法。

【発明の詳細な説明】

### 【0001】

【発明の属する技術分野】本発明は、周辺のハードウェアをシミュレーションしてハードウェアを検証するハードウェア開発環境と、主にCPU部分のハードをシミュレーションしてソフトウェアを検証するソフトウェア開発環境をリンクして、ソフトウェアとハードウェアの両者を並行して開発を進める協調シミュレーション方法に関する。

### 【0002】

【従来の技術】近年のCPUを内部に持つLSI(ASIC)の開発期間の短縮に伴い、ハードウェア実行デバイス製造以前のシステム設計段階でのハードウェアとそのハードウェアで動作するソフトウェアを含む機能検証が要求されている。

【0003】このため、周辺のハードウェアをシミュレーションしてハードウェアを検証するハードウェア開発環境と、主にCPU(以下、ターゲットCPUという)部分のハードをシミュレーションしてソフトウェアを検証するソフトウェア開発環境をリンクして、ソフトウェアとハードウェアの両者を並行して開発を進める協調シミュレーションを行う方法が提案されている。

【0004】ハードウェア開発環境では、主に周辺のハードウェアを、HDL(Hardware Description Language)を用いてシミュレーションしてシステム動作の検証を行っており、ソフトウェア開発環境では、ターゲットCPU用に開発されたソフトウェア(以下、ターゲットプログラムという)をコンパイルし、ターゲットCPU用のオブジェクトコードをシミュレータのメモリモデルに与えてシミュレーションしてターゲットプログラムの検証を行っている。

【0005】ソフトウェア開発環境でのシミュレーション方法には、HDLを用いて擬似的にターゲットCPUのCPUモデルを作成し、ターゲットプログラム(ターゲットCPU用のソースプログラム)をコンパイルしてターゲットCPU用のオブジェクトコードを生成し、ターゲットCPUの命令セットでシミュレーションを行う方法がある。このHDLを用いた方法では、ターゲットCPUのハードそのものを再現するので正確にシミュレーションを行うことができるが、クロック単位までシミュレーションを行うのでシミュレーションに非常に時間がかかるという特徴がある。

【0006】また、ソフトウェア開発環境でのシミュレーション方法には、ターゲットCPUの命令レベルのCPUモデルをC言語で作成し、ターゲットプログラムをコンパイルしてターゲットCPU用のオブジェクトコードを生成し、ターゲットCPUの命令セットでシミュレーションを行う方法がある。この方法は、クロック単位までシミュレーションしていないので、シミュレーション

ンがやや不正確ではあるが、前述のHDLを用いる方法よりは、シミュレーション時間が高速である。

【0007】このような命令レベルでシミュレーションを行う方法を用いたものに特開平6-250874号公報、特開平6-324883号公報に記載の発明がある。これらは、いずれもCPU以外の周辺デバイスの動作に係る部分のみ、ハードウェアの論理シミュレーションで実行し、その他のソフトウェア部分をホストCPU上でCPUモデルを用いてシミュレーションしている。

【0008】図4は、従来の命令レベルでシミュレーションを行う方法を概略的に示したものであり、ターゲットプログラムをすべて命令レベルのシミュレータであるISS(Instruction Set Simulator)上で実行する構成を示している。ターゲットプログラムには、アプリケーション、デバイスドライバ、OSが含まれる。

【0009】CPUを内部に持つLSIのシミュレーションを行う際、ターゲットプログラムは、すべてターゲットCPU向けにコンパイルし、ISSを介してホスト上で実行し、他のハードウェア部分は従来の論理シミュレータを用いる方法が採られている。CPUのみアクセスするメモリは、ISS側にローカルに持たせている。

【0010】さらに、ソフトウェア開発環境でのシミュレーション方法には、CPUモデルを使用しないで、ターゲットプログラムをコンパイルしてホストCPU用のオブジェクトコードを生成し、直接ホストCPUの命令セットでシミュレーションを行う方法がある。この方法は、シミュレーションが非常に高速ではあるが、クロックレベルまでシミュレーションできないことはもちろん、ハードウェアに依存する部分、インターフェース部分をシミュレーションできないので、シミュレーションが不正確である。

【0011】このような直接ホストCPUの命令セットでシミュレーションを行う方法を用いたものに特開平10-260865号公報に記載の発明がある。この発明は、CPU以外の周辺デバイスの動作に係る部分のみ、ハードウェアの論理シミュレーションで実行し、他のソフトウェア部分をホストCPU上でシミュレーションして高速化を図っている。

### 【0012】

【発明が解決しようとする課題】従来は、シミュレーションを高速で行うためには、ターゲットプログラムの実行を、ISSを用いて行うか、シミュレータを実行するホストマシンネイティブな環境で行うかのどちらかであった。前者の場合は、LSI(ASIC)に載せるCPUモデルの上で実行するために、実際に近い環境でターゲットプログラムの開発を行うことができ、対象とするCPUの命令レベルのデバッグの可能となるが、エミュレートしているISS上でターゲットプログラムを実行するために、速度的にはHDLを用いて実行する場合よ

りは優れているが、プロセッサネイティブな環境で実行する場合に比べると劣るという問題があった。

【0013】一方、後者の場合は、ターゲットプログラムの実行をホストマシンネイティブな環境で行うので、高速ではあるが、CPU固有の問題をシミュレートできないという問題があった。

【0014】本発明の目的は、高速かつ正確なシミュレーションを実現するソフトウェア／ハードウェア協調シミュレーション方法を提供することにある。

### 【0015】

【課題を解決するための手段】本発明のソフトウェア／ハードウェア協調シミュレーション方法は、CPU以外のハードウェアはハードウェア記述言語を用いてシミュレーションを行い、ハードウェアを直接制御するソフトウェアはCPUシミュレータを用いてシミュレーションを行い、ハードウェアを直接制御しないソフトウェアはホストネイティブでシミュレーションを行うことを特徴とする。

【0016】また、前記CPUのみアクセスするメモリは、前記CPUシミュレータ側にローカルに持つことを特徴とする。

### 【0017】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して説明する。

【0018】図1は、本発明のソフトウェア／ハードウェア協調シミュレーション方法を説明する図である。

【0019】図1に示すハードウェア開発環境2では、ターゲットCPU以外の部分のLSI(ASIC)をシミュレーションして検証し、ソフトウェア開発環境1では、ターゲットCPU部分のハードを、命令レベルのCPUのシミュレータであるISS(Instruction Set Simulator)でシミュレーションしてソフトウェアを検証する。このISSは、一般的にはC言語で記述されており、RTL記述に比べてシミュレーション時間は数10～100倍程度速い。

【0020】ASIC-ISSインタフェース4は、CPUシミュレータであるISSとLSI(ASIC)とをインタフェースしており、ISSは、ASIC-ISSインタフェース4を介してASICから割り込み命令等を受け取る。ターゲットプログラム3には、アプリケーション、デバイスドライバ、OSが含まれる。

【0021】また、CPUのみアクセスするメモリは、ハードウェア記述で持たせるのではなく、ローカルに持たせることによりメモリアクセスの高速化を図っている。ターゲットCPUのみアクセスするメモリがハードウェアにすると、ハードウェアのクロック単位のシミュレーションに影響されて処理に時間がかかるてしまうために、ソフトウェア環境に擬似的にメモリを持たせている。

【0022】協調シミュレーション環境として、ターゲ

ットCPU以外のハードウェアは、RTL記述で論理シミュレータを用い、ターゲットCPUは、ISSを用い、ISSのシミュレータ、デバッガを用いる。

【0023】ここで、ターゲットCPUで実行するプログラムのうち、ハードウェアを直接操作する部分であるOSやデバイスドライバ等は、ターゲットCPU向けにコンパイルし、ISSを介してホスト上で実行し、ハードウェアとは無関係に実行されるアプリケーションは、ハードウェア依存がないことからシミュレーション精度が要求されないので、ホストマシンネイティブでコンパイルし、ホスト上で実行する。

【0024】具体的に説明すると、OSやデバイスドライバ等のターゲットプログラム（ターゲットCPU用の例えばC言語で記述されたソースプログラム）を、ターゲットCPUのコンパイラでコンパイルして、ターゲットCPU用のオブジェクトコードを生成し、ハードウェアとは無関係に実行されるアプリケーション（例えば、メールソフト）を、ホストCPUのコンパイラでコンパイルして、ホストCPU用のオブジェクトコードを生成する。

【0025】ターゲットCPU用のオブジェクトコードとホストCPU用のオブジェクトコードをリンクするときに、ISSで走らせるコードのアドレスを出力させ、ISSは、オブジェクトを実行するときに、コードのアドレスによって、実行環境をISSとホストCPUとに区別する。

【0026】上述のように、本発明は、ハードウェアを直接制御しないターゲットプログラムを、ホストCPUのオブジェクトコードへコンパイルした後にホストCPU上で直接実行するので、（アプリケーション、デバイスドライバ、OS等の）全てのプログラムを、ISSを介して実行する場合に比べて、高速にシミュレーションを行うことができる。

【0027】また、ハードウェアを直接制御するターゲットプログラムをISSを介して実行するので、ターゲットプログラムのすべてをホストマシンネイティブな環境で実行する場合に比べて、ハードウェアに依存する事象も忠実にシミュレーションを行うことができる。

【0028】図2は、本発明のソフトウェア構成を説明する図である。ホストマシンネイティブでコンパイル、実行した場合には、ターゲットCPUの命令レベルでのシミュレーションは不可能となるため、この差異を埋めるアプリケーションの処理時間補正部をアプリケーションとホストマシンネイティブな環境との間に実装している。このアプリケーションの処理時間補正部では、ホストマシンネイティブな環境を用いた場合とターゲットCPUの命令セットを使った場合の処理時間の差などを統計的手法により求め、アプリケーションの処理時間の補正を行うものである。

【0029】図3は、ハードウェアを直接制御しないア

プリケーションが電子メールソフトである場合の協調シミュレーションを説明する図である。

【0030】図3に示すように、パケットをイーサネットから受信してメモリへ格納し、受信したことを割り込みで通知するハードウェア部分は、RTL記述で論理シミュレータを用いてシミュレーションを行う。

【0031】ハードウェアの格納方式からTCP/IPモジュールの格納方式に変換してTCP/IPモジュールを起動するデバイスドライバ部分は、ISSを介して実行し、シミュレーションを行い、ヘッダを解釈して電子メールソフトを実行する。ハードウェアに依存しない部分は、ホストCPU上で直接実行してシミュレーションを行う。

【0032】

【実施例】次に、本発明の実施例について図面を参照して説明する。

【0033】図4は、本発明の一実施例の構成を示すブロック図である。本実施例の協調シミュレーション方式は、ソフトウェアシミュレータ10と、ハードウェア論理シミュレータ11と、データ送受信、実行シミュレータの切り替え、全体のクロックの管理を行うシミュレータ管理手段12とから構成される。

【0034】また、ソフトウェアシミュレータ1.0は、実行するプログラム13のプログラムカウンタを比較し、プログラムの実行環境をホストマシンにするか、命令レベルシミュレータにするかを判断するプログラムカウンタ比較手段14と、この比較手段14の結果によって、実行環境を切り替える命令実行環境切り替え手段15と、命令レベルシミュレータ16と、命令レベルシミュレータ16を介さない場合に実行時間の補正を行う実行時間補正手段17と、命令実行手段18と、ハードウェア論理シミュレータ11とデータをやり取りするデータ転送手段19およびデータ受信手段20とから構成される。

【0035】また、ハードウェア論理シミュレータ11は、ソフトウェアシミュレータ10とデータをやり取りするデータ転送手段20およびデータ受信手段21と、命令実行手段23とから構成される。

【0036】図5は、本発明において、実行オブジェクトおよびプログラムカウンタ比較手段を作成する手順の一例である。

【0037】デバッグ対象となるソフトウェアはコンパイルし、OS等とともにリンクして、メモリに格納する実行オブジェクトを作成する。このコンパイル、リンク時にオブジェクトコードのアドレス情報と処理内容の対応表を作成する。その後、この対応表より処理内容によって、アドレス情報と命令実行環境（命令レベルシミュレータ環境で実行するべき命令：ターゲットCPUで実際に動作した時の実行タイミングまで正確にシミュレーションするべき命令（例えば、ハードウェア直接制御命

令)、(ホストCPUで)直接命令を実行する環境で実行するべき命令:ターゲットCPUでの実動作タイミングを考慮する必要のない命令(例えば、ハードウェアを直接制御しない命令)とのマッピングを行い命令実行環境を記述した対応表を完成する。プログラムカウンタ比較手段においては、この対応表を用いる。

【0038】次に、図4を用いて、本実施例の動作を説明する。まず、プログラムの実行を開始する。プログラムカウンタ比較手段14において、プログラムカウンタ値から命令実行環境を読み出す。プログラムカウンタ値がハードウェア論理シミュレータ11へのアクセスを示している場合には、データ転送手段19を介してデータをハードウェア論理シミュレータ11へ送信する。ハードウェア論理シミュレータ11では、データ受信手段21でデータを受信することをトリガとして、命令実行を行う。

【0039】また、プログラムカウンタ比較手段14において、プログラムカウンタ値がCPU内部での実行を示しており、かつ、命令レベルシミュレータで実行するべき命令であることを示している場合、その命令は命令レベルシミュレータに渡され、内部で処理される。

【0040】また、プログラムカウンタ比較手段14において、プログラムカウンタ値がCPU内部での実行を示しており、かつ、直接命令実行環境で実行するべき命令であることを示している場合、まず、実行時間補正手段17に渡され、NOP命令を補正時間分だけ挿入されたのち、命令実行手段18で命令を実行する。

【0041】以上、本発明の一実施例を述べてきたが、直接ホストマシンで実行するソフトウェアと命令レベルシミュレータ上で実行するプログラムの分類として、例えば、命令レベルシミュレータ上では周辺デバイスを直接制御や読み書きするデバイスドライバ、OSなどが考えられる。

#### 【0042】

【発明の効果】以上説明したように、本発明は、CPUを内部に持つLSIのシミュレーションを行う際、ソフトウェアの実行環境をハードウェアに近い部分はISS

を用い、CPU上で走るアプリケーションはホストマシンネイティブな環境を用いることによって、高速なシミュレーションを行うことができる。

【0043】また、ハードウェアとのインターフェース部はISSを用いているため、ソフトウェアすべてをホストマシンネイティブな環境で実行するのに比べて、ハードウェアに依存する事象も忠実にシミュレートすることができる。

#### 【図面の簡単な説明】

【図1】本発明のソフトウェア／ハードウェア協調シミュレーション方法を説明する図である。

【図2】本発明のソフトウェア構成を説明する図である。

【図3】アプリケーションが電子メールソフトである場合の協調シミュレーションを説明する図である。

【図4】本発明の一実施例の構成を示すブロック図である。

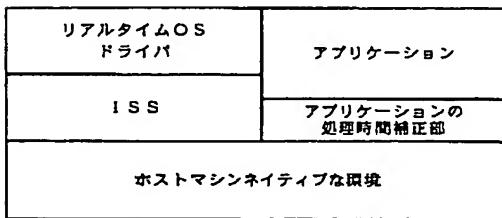
【図5】実行オブジェクトおよびプログラムカウンタ比較手段を作成する手順の一例を示す図である。

【図6】従来の協調シミュレーション方法を説明する図である。

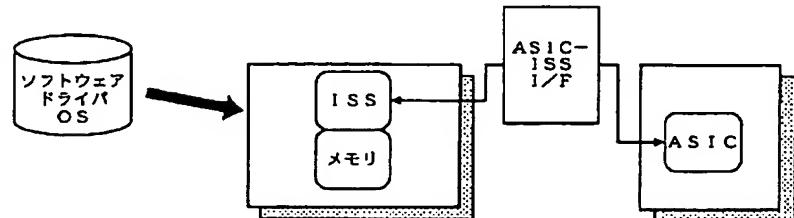
#### 【符号の説明】

- 1 ソフトウェア開発環境
- 2 ハードウェア開発環境
- 3 ターゲットプログラム
- 4 ASIC-ISSインターフェース
- 10 ソフトウェアシミュレータ
- 11 ハードウェア論理シミュレータ
- 12 シミュレータ管理手段
- 13 実行プログラム
- 14 プログラムカウンタ比較手段
- 15 命令実行環境切り替え手段
- 16 命令レベルシミュレータ
- 17 実行時間補正手段
- 18, 23 命令実行手段
- 19, 22 データ転送手段
- 20, 21 データ受信手段

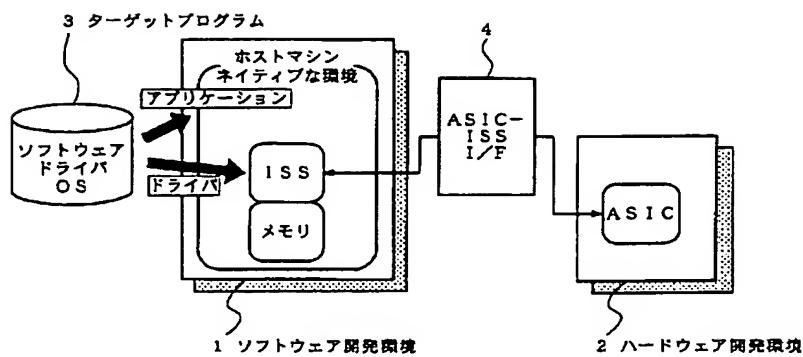
【図2】



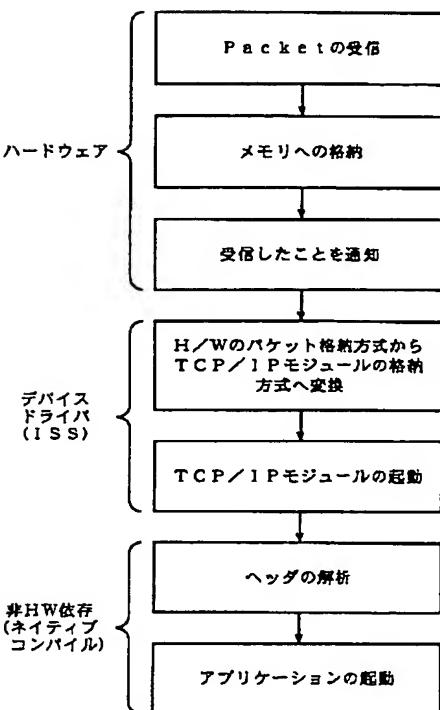
【図6】



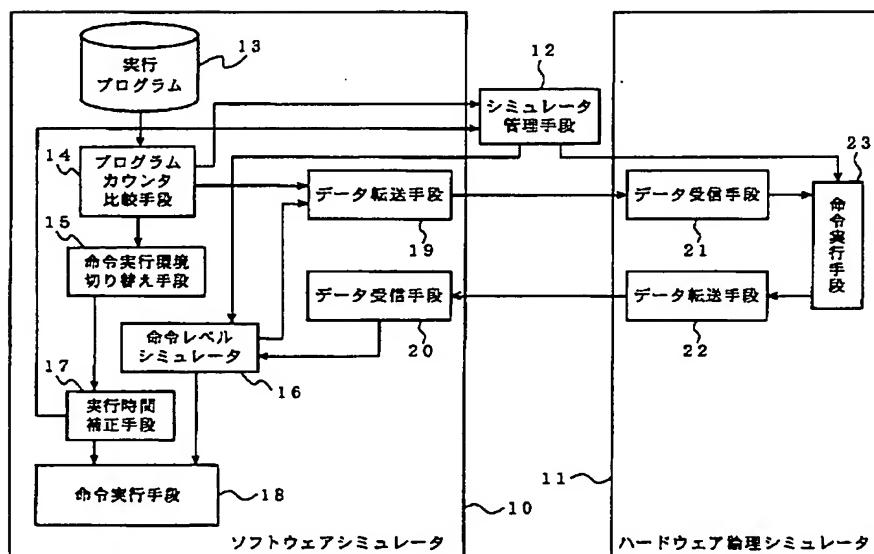
【図1】



【図3】



【図4】



【図5】

